



SOFTWARE EN MATLAB PARA UN SISTEMA DE ADQUISICIÓN DE DATOS UTILIZANDO LA TARJETA DE SONIDO DE UNA PC.

César Omar Jiménez Tintaya^{a,b*}

^aFacultad de Ciencias Físicas, Universidad Nacional Mayor de San Marcos, Ap. Postal 14-0149, Lima 14, Perú.

^bFacultad de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Ingeniería, Lima-Perú.

Resumen

Dado el alto costo de las tarjetas de adquisición de datos, se ha diseñado e implementado un sistema de adquisición de datos mediante la tarjeta de sonido de la PC, el cual es un conversor analógico digital de 2 canales, con una resolución de 16 bits y una frecuencia de muestreo de 1Hz hasta 44.1 kHz. El costo es relativamente bajo o prácticamente nulo, debido a que la mayoría de las PC actuales tienen una tarjeta de sonido. La interfaz gráfica de usuario ha sido implementada en Matlab, con opciones para seleccionar el número de canales, número de bits, frecuencia de muestreo y tiempo de grabación. La señal es graficada tanto en el dominio del tiempo como en el dominio de la frecuencia vía su transformada de Fourier. También se incluye una opción para realizar el filtrado de la señal, reproducir la señal en caso de trabajar con frecuencias de audio y otras utilidades. Este sistema puede aplicarse al procesamiento digital de señales de audio, ciertos experimentos de laboratorio, adquisición de señales sísmicas, adquisición de señales biomédicas, osciloscopio de baja frecuencia, etc. Se discuten las ventajas y desventajas del sistema de adquisición.

Palabras claves: software, matlab, adquisición de datos.

Abstract

Given the high cost of cards of data acquisition, it has been designed and implemented a system of data acquisition by means of the sound card of the PC, which is a digital analogical converter of 2 channels, with a resolution of 16 bits and one frequency of sampling of 1Hz up to 44,1 kHz. The cost is relatively low or practically null, because most of the PC present they have a sound card. The graphical interface of user has been implemented in Matlab, with options to select the number of channels, I number of bits, frequency of sampling and time of recording. The signal is graficada as much in the dominion of the time like in the dominion of the frequency via transformed his of Fourier. Also an option is included to make the filtrate of the signal, to reproduce the signal in case of working with frequencies of audio and other utilities. This system can be applied to the digital signal processing of audio, certain experiments of laboratory, acquisition of seismic signals, acquisition of biomedical signals, oscilloscope of LF, etc. To the advantages and disadvantages of the acquisition system are discussed.

PACS: 01.50.hv; 07.05.Hd; 07.05.Rm

keywords: software, matlab, data acquisition.

1. Introducción

Explicaremos algunos conceptos y términos de la tarjeta de sonido. Para comprender las tarjetas de sonido, es necesario conocer varios conceptos y términos. Los términos como 16 bits, calidad de CD, etc. Son sólo algunos. Los conceptos como el muestreo y la DAC (conversión digital a analógico) a menudo están diseminados en las descripciones de

los productos de sonido de las PC multimedia actuales.

La naturaleza del sonido

Cada sonido es producido por vibraciones que comprimen aire u otras sustancias. Estas ondas de sonido viajan en todas direcciones, expandiéndose de forma similar a un globo desde la fuente del sonido.

* Corresponding author. e-mail: cjimenez@axil.igp.gob.pe

Cuando estas ondas llegan al oído, provocan las vibraciones que percibimos como sonido.

Las dos propiedades básicas de todo sonido son la frecuencia e intensidad. La frecuencia es simplemente la velocidad a la que se producen las vibraciones. Se mide en hertz (Hz) o ciclos por segundo. Un ciclo es una vibración completa. La cantidad de hertz es la frecuencia; entre mas alta sea, mayor será el tono. No es posible escuchar en todas las frecuencias posibles. Muy pocas personas pueden oír menos de 20Hz o más de 20kHz. De hecho la nota mas baja en un piano tiene una frecuencia de 27Hz y la más alta un poco mas de 4kHz. Una estación de radio FM (frecuencia modulada) transmite notas de hasta 15 kHz. A la intensidad del sonido se le denomina amplitud (está asociado con el volumen del sonido). Esta intensidad depende de la fuerza de las vibraciones que producen el sonido. Por ejemplo, una cuerda de piano vibra levemente cuando la tecla se pulsa con suavidad. La cuerda oscila de arriba hacia abajo en un arco angosto y el tono que emite es suave. Sin embargo, si la tecla se pulsa con fuerza, la cuerda oscila en un arco mas amplio. El volumen de los sonidos se mide en decibeles(db). El susurro de las hojas secas está clasificado en 20db, el ruido promedio en la calle es 70db y un trueno cercano en 120db.

Respuesta en frecuencia

La calidad de una tarjeta de sonido a menudo se mide por dos criterios: respuesta en frecuencia (o rango) y distorsión armónica total.

La respuesta en frecuencia de una tarjeta de sonido es el rango en el que un sistema de audio puede grabar o reproducir un nivel de amplitud constante y audible. Muchas tarjetas manejan de 30Hz a 20kHz. Entre mas amplia sea la separación, la tarjeta de sonido será mejor. La distorsión armónica total mide la linealidad de una tarjeta de sonido, la rectitud de una curva de respuesta en frecuencia. La distorsión armónica total es una medida de precisión en la reproducción del sonido. Todo elemento no lineal causa distorsión en la forma de armonía. Entre mas bajo sea el porcentaje de distorsión, es mejor.

Muestreo

Con una tarjeta de sonido, una PC puede grabar una forma de onda de audio. La forma de onda de audio (también conocido como sonido en muestras o digitalizado) utiliza a la PC como si fuera una grabadora de cinta. Integrados dentro de una tarjeta de sonido existen pequeños chips denominados ADC (convertidores analógico a digital), que convierten ondas de sonido en bits digitales que la computadora puede entender. En forma similar, los convertidores DAC (digital a analógico) transforman los sonidos

grabados en la PC en algo audible a través de los parlantes.

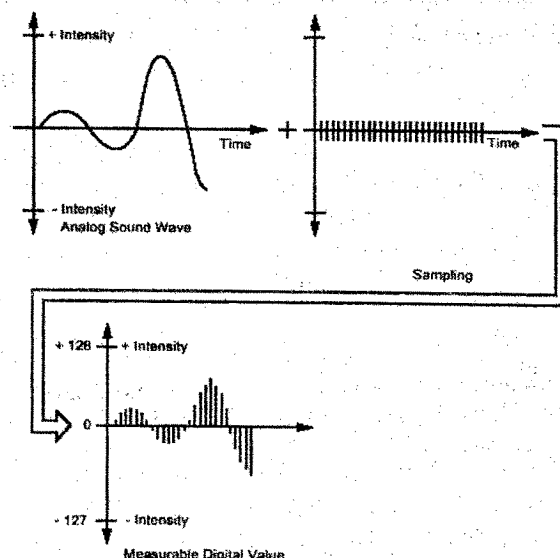


Fig. 1. El proceso del muestreo

El muestreo es el proceso de convertir las ondas originales de sonido analógico en señales digitales que puedan almacenarse y reproducirse después. Se toman instantáneas de los sonidos analógicos y se almacenan. Por ejemplo, se podría medir el sonido en el momento t , con una amplitud " y ". Entre mas alta sea la tasa de la muestra, el sonido digital es más preciso con respecto a la fuente original. El teorema del muestreo establece que si una señal se muestrea de manera que se cumpla la condición $w_s > 2w_m$, es decir, que la frecuencia de muestreo tiene que ser mayor que el doble de la máxima frecuencia contenida en la señal, entonces es posible recuperar la señal.

La resolución en bits

Las especificaciones originales de una PC multimedia requerían un sonido de 8 bits. Esto no significa que la tarjeta de sonido deba ajustarse a una ranura de expansión de 8 o 16 bits. Mas bien, el audio de 8 bits significa que la tarjeta de sonido emplea 8 bits para representar cada muestra de sonido. Esto se traduce en 256 valores digitales posibles para representar cada muestra (menos calidad que los

65,536 valores posibles con una tarjeta de sonido de 16 bits). En general, una tarjeta de 8 bits es adecuada para la voz grabada, mientras que el sonido de 16 bits es mejor para cubrir la demanda de música.

La Fig. 2 muestra la diferencia entre el sonido de 8 y 16 bits. Hoy en día, no se recomendaría nada inferior a una tarjeta de 16 bits, la cual ofrece una muy alta resolución.

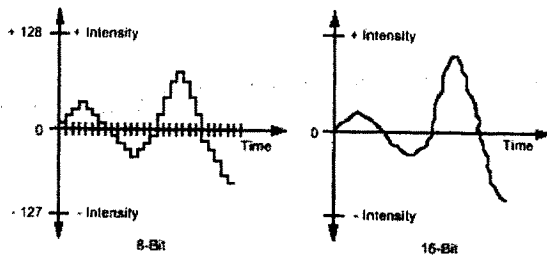


Fig. 2. Resolución de la señal

Además de la resolución, la razón de muestreo o frecuencia determina que tan a menudo la tarjeta de sonido mide el nivel de sonido que se está grabando o reproduciendo. Básicamente, como ya se mencionó, debe realizar el muestreo a casi dos veces la frecuencia mas alta que desee reproducir, mas 10 por ciento adicional para dejar fuera señales no deseadas. Los humanos pueden escuchar hasta 20 kHz. Entonces la frecuencia de muestreo debe ser de 44.1 kHz, la misma que emplean los CDs de audio de alta fidelidad.

El sonido grabado a 11kHz es más confuso que el que se muestrea a 22 kHz. Un sonido con una muestreo de 44 kHz (calidad de CD de audio) en estereo de 16 bits (dos canales) requiere 10.5Mb de espacio en disco por minuto. La misma muestra de sonido en 8 bits monofónico

(un solo canal) a 11 kHz ocupa una decimosexta parte de ese espacio.

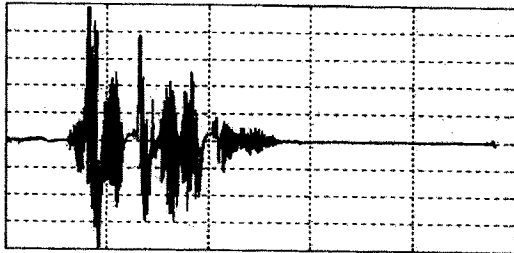


Fig. 3. Señal de audio en el dominio del tiempo

El Lenguaje Matlab

Matlab (Matrix Laboratory) es un lenguaje de programación técnico-científico que básicamente trabaja con variables vectoriales y matriciales. Es fácil de utilizar debido a que contiene varias cajas de herramientas con funciones incorporadas (toolbox de procesamiento de señales, teoría de control, wavelets, matemática simbólica, etc). La versión que hemos utilizado es la 6.5. Es posible trabajar en programación estructurada o en programación orientada a objetos con una interfaz gráfica de

usuario. Es casi un estándar en la programación en ciencias e ingeniería. Es apropiado para el caso de muchas señales de interés, donde la frecuencia de muestreo sea menor que 44.1 kHz. Ejemplo de programa en Matlab:

```
load señal.txt % carga el archivo de datos
y = señal; % declara una variable
N = length(y); % longitud del vector "y"
Fs = 8000; % frecuencia de muestreo
t=(0:N-1)/Fs; % declara el tiempo
plot(t,y), grid % gráfica de la señal
```

El resultado de este pequeño segmento de programa es la Fig. 3. Solo se han utilizado 6 líneas. Si esto se realizara en otro lenguaje como Pascal o C++, el programa sería mucho mas largo. La potencia de Matlab radica en su simplicidad. El inconveniente de Matlab es que se torna lento si la longitud del vector (o matriz) es relativamente grande. Se debe evitar el uso del bucle "for k=1:N" para N muy grande, debido a que ralentiza el cálculo, el bucle *for* es extremadamente ineficiente en Matlab, ya que se trata de un lenguaje interpretado. Sin embargo, se puede vectorizar las operaciones de cálculo, lo que es mas rápido. El proceso de convertir un bucle *for* en una operación matricial o vectorial podría denominarse vectorización.

Herramientas Matemáticas

La transformada de Fourier es una herramienta matemática muy importante en el procesamiento de señales que sirve para transformar una señal del dominio del tiempo al dominio de la frecuencia. Para el caso continuo tenemos:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (2)$$

y para el caso discreto:

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi kn/N} \quad (3)$$

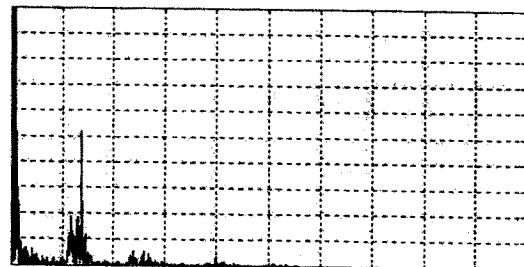


Fig. 4. Señal en el dominio de la frecuencia

El espectro de frecuencias de una señal nos permitirá saber qué componentes frecuenciales están presentes. Esto servirá para aplicar un filtro adecuado si fuera necesario. El código en Matlab es el siguiente:

```
Y = fft(y,N); % N: longitud del vector y
Pyy = Y.*conj(Y) / N; f = (Fs)*(0:N/2-1)/ N;
plot(f,Pyy(1:N/2)), grid
```

El proceso del filtrado consiste en eliminar, atenuar o amplificar ciertas componentes frecuenciales de una señal. Por ejemplo, la Fig. 5 muestra una señal contaminada con ruido de baja frecuencia, la cual debe ser tratada con un filtro pasa-alta para eliminar dicha componente frecuencial. El resultado es la señal de la Fig. 6. Según su aplicación, un filtro puede ser clasificado como pasabajo, pasa-alto, pasa banda o elimina banda. Un tipo de filtro comúnmente utilizado es el de Butterworth, debido a su pequeña banda de transición y a la ausencia de ripple en la banda de paso. Dependiendo del tipo de señal, se escogerá la frecuencia de corte del filtro. Para señales de audio, $f_{corte}=100\text{Hz}$. Ejemplo de un filtro pasa alta en Matlab:

```
f_corte=100; %elimina debajo de 100Hz
[b, a] = butter(5,f_corte/(Fs/2),'high');
yf = filtfilt(b,a,y);
plot(t,y,'red'), grid on, zoom xon
```

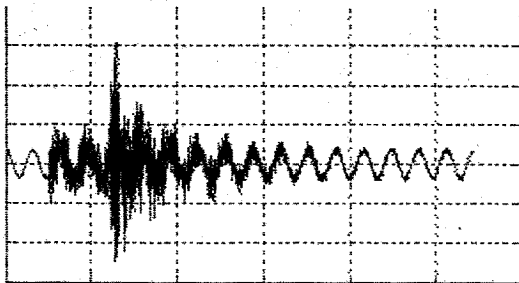


Fig. 5. Señal contaminada con ruido

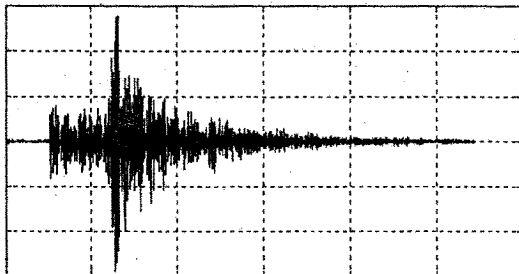


Fig. 6. Señal anterior después de filtrar

2. Procedimiento y Método.

Se utilizarán algunas funciones, las cuales vienen incorporadas en Matlab para realizar la adquisición de datos, las cuales serán descritas a continuación:

Comando audiorecorder

Es un objeto de grabación de audio en Windows. Crea un objeto por defecto de 8,000 Hz de muestreo, monofónico (1 canal), y 8 bits de resolución. Audiorecorder (Fs, Nbits, Nchans) crea un objeto con una frecuencia de muestreo de "Fs" (la cual, en principio, podemos variar de 1Hz hasta 44kHz), resolución de "Nbits" (Nbits puede ser 8, 16 o 24, dependiendo de la tarjeta de sonido) y de "Nchans" canales. Este último solo puede tomar 2 valores: 1 (monofónico) o 2 (estéreo). Estos parámetros pueden variarse dentro de su respectivo rango, lo que dependerá de la aplicación que deseemos realizar.

Comando record

Este comando graba los datos y la información del evento a un archivo. Record (obj) alterna la propiedad de estado de grabación del objeto (Obj) entre ON y OFF. Cuando el estado de grabación está en ON, los comandos escritos al dispositivo o los datos leídos del dispositivo serán grabados en el archivo especificado. El archivo de grabación está en formato Ascii.

Comando getaudiodata

Este comando nos permite cargar la señal en el espacio de trabajo (work space) de Matlab, mediante un vector o matriz especificado. El formato del vector puede ser de tipo entero o real. Ejemplo:

```
y = getaudiodata(r, 'int16');
% trae la data como un vector "int16"
y = getaudiodata(r);
% trae la data como un vector "real"
```

Estos tres comandos son el núcleo del software que habremos de implementar para el sistema de adquisición de datos. Una vez obtenida la señal, se procederá a analizarla. Para esto se pueden emplear las "herramientas matemáticas" descritas anteriormente: la transformada de Fourier para obtener la representación de la señal en el dominio de la frecuencia y el filtrado de la señal para eliminar las componentes frecuenciales no deseadas.

La interfaz gráfica de usuario (GUI), el cual va a permitir al usuario interactuar con el ordenador de

una manera rápida en la solución de problemas, se realiza con el comando "guide", el cual crea un Fig-file, en el cual pueden insertarse botones, ejes, deslizadores, texto de edición, texto estático, menús, etc. Cada uno de los cuales tiene ciertas propiedades y características y realiza una función determinada dentro de la programación orientada a objetos (callback function). Por ejemplo: para filtrar la señal, simplemente se hace click en el botón <Filtro>.

3. Resultados

Se realizó la ejecución del programa utilizando la versión 6.5 de Matlab. El software es sencillo de utilizar, es completamente interactivo. Se utilizó un generador de señales para analizar la respuesta del sistema ante diferentes tipos de señales.

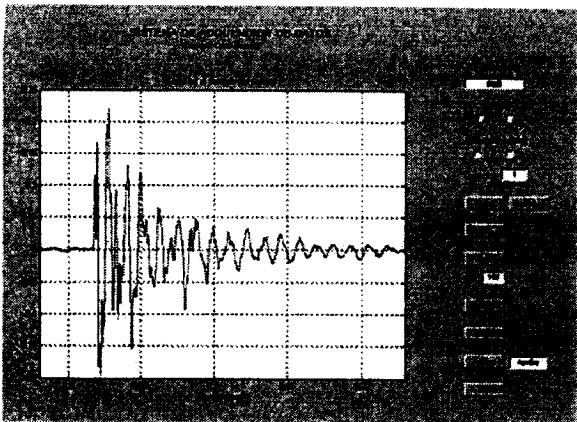


Fig. 7. Interfaz gráfica de usuario

La Fig. 7 muestra una señal grabada a 8000 Hz de muestreo, con una resolución de 16 bits, con un tiempo de grabación de 1 seg.

4. Discusión y Conclusiones

Se ha diseñado e implementado un método de bajo costo para la adquisición de señales.

El software es factible de ser mejorado o adecuado a la necesidad del usuario. Puede ser software libre.

La tarjeta de sonido ha sido diseñado para trabajar con señales de audio. Sin embargo, podemos trabajar con otros tipos de señales.

La amplitud de la señal de entrada debe ser del

orden de los milivoltios. No se debe intentar trabajar con señales de la red eléctrica.

Es necesario realizar un filtrado para eliminar el ruido de la red de alimentación (60 Hz).

La función de monitoreo en tiempo real realiza la grabación de datos de segundo en segundo y la señal se ve como en cámara lenta. Necesita ser mejorado.

El presente software es ideal para realizar trabajos de procesamiento digital de señales de voz. Podrían implementarse mas funciones en Matlab como análisis espectral, cálculo del coeficiente de correlación, etc.

Código Fuente Núcleo

```

Fs = 8000; % frecuencia de muestreo
Nbit = 16; % Numero de bits
Nchan = 1; % numero de canales
tg = 3; % Tiempo de grabacion r =
audiorecorder(Fs, Nbit, Nchan); record(r);
pause(tg);
stop(r);
y = getaudiodata(r);
y = y - mean(y); N = length(y);
[A,I] = max(abs(y));
% Aplicando un filtro pasa-alta
%fcorte = 100;
%[b, a] = butter (5,fcorte/(Fs/2), 'high');
%yf = filtfilt (b,a,y);
%[Af,I] = max(abs(yf));
%y = (A/Af)*yf;
t = (1:N)/Fs;
subplot(2,1,1), plot(t,y), grid, zoom xon title('Señal
en el dominio del tiempo') xlabel('Tiempo (seg)')
Y = fft(y,N);
Pyy = Y.*conj(Y) / N;
f = (Fs)*(0:N/2-1)/ N;
subplot(2,1,2), plot(f,Pyy(1:N/2)), grid, zoom xon
title('Espectro de frecuencias de la señal')
xlabel('Frecuencia (Hz)')

```

Referencias

- [1]. Herón Morales M. "Matlab 7, métodos numéricos", Grupo Editorial Magabyte, Lima 2005
- [2]. Scott Mueller. "Manual de actualización de PCs", Prentice Hall, Mexico 1999
- [3]. Building GUIs with Matlab. The Mathworks Inc., June 1997